



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/608,204	06/27/2003	Roberto J. Bayardo JR.	END920030039	7637
23550	7590	10/30/2008		
HOFFMAN WARNICK LLC			EXAMINER	
75 STATE STREET			MITCHELL, JASON D	
14TH FLOOR			ART UNIT	PAPER NUMBER
ALBANY, NY 12207			2193	
			NOTIFICATION DATE	DELIVERY MODE
			10/30/2008	ELECTRONIC

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PTOCommunications@hoffmanwarnick.com

<b>Office Action Summary</b>	<b>Application No.</b> 10/608,204	<b>Applicant(s)</b> BAYARDO ET AL.
	<b>Examiner</b> Jason Mitchell	<b>Art Unit</b> 2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If no period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### **Status**

- 1) Responsive to communication(s) filed on 20 June 2008.
- 2a) This action is FINAL.      2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### **Disposition of Claims**

- 4) Claim(s) 10-12,14-17,19,23,24,32,33,36 and 37 is/are pending in the application.
  - 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 10-12,14-17,19,23-24,32-33,36-37 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### **Application Papers**

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.
 

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### **Priority under 35 U.S.C. § 119**

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) All    b) Some \* c) None of:
    1. Certified copies of the priority documents have been received.
    2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
    3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### **Attachment(s)**

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) Notice of Informal Patent Application
- 6) Other: \_\_\_\_\_

### **DETAILED ACTION**

This action is in response to an amendment filed on 6/20/08.

Claims 10-12, 14-17, 19, 23, 24, 32-33 and 36-37 are pending in this application.

#### ***Response to Arguments***

**Applicant's arguments filed 6/20/08 have been fully considered but they are not persuasive.**

In the par. bridging pp. 12-13 the applicants state:

In the Office Action, claims 10-14, 23 and 32-33 are rejected under 35 USC 101 as allegedly being directed to non-statutory subject matter. Without conceding the correctness of the Office's interpretation, and to facilitate early allowance of the pending claims, claims 10, 23 and 32 have been amended to include "at least one computer device." Accordingly, Applicant asserts that the bases for the Office's rejection have been obviated and respectfully request withdrawal of the rejection.

The examiner notes that a 'device' does not necessarily indicate a physical component as is required. Specifically, a 'device' can refer to a plan or scheme. The example that comes to mind is a 'pneumonic device' which is only an abstract idea used to aid memory. This type of rejection is commonly overcome by the inclusion of a known hardware component (e.g. a processor or memory).

In the last full par. on pg .13, the applicants state:

A review of the four references cited by the Office in alleging obviousness (to wit: Verbeke, Greenfeld, Kataoka, and Ekkel), reveals that none of the references teach or suggest "an Integrated Development Environment." None of the references teach or suggest an IDE that include "classification-based navigation and content dependent searching for code patterns."

(emphasis in original)

The examiner respectfully disagrees. For example in par. [0189] Verbeke discloses "Services ... may be used to facilitate application development". Those of ordinary skill in the art would have recognized this as indicating a development environment which is integrated, at least, with other peer nodes through the navigation and searching disclosed in e.g. par. [0180]-[0181].

Further, the applicants are reminded that reliance on a large number of references in a rejection does not, without more, weigh against the obviousness of the claimed invention. *In re Gorman*, 933 F.2d 982, 18 USPQ2d 1885 (Fed. Cir. 1991).

In the first full par. on pg. 14, the applicants state:

The referenced sentence in its entirety merely states "A peer group is a collection of peers connected by a network that share a common set of interests and that have agreed upon a common set of rules to publish, share and access any computer content (code, data, applications, or other collections of computer representable resources), and communicate among themselves." This reference does not teach any of the cited elements of the claimed invention. Verbeke does not teach anything about "a code pattern classifier for analyzing source code" or "a code pattern requestor for retrieving the source code." The entirety of Verbeke fails to teach these elements. Greenfeld, Kataoka, and Ekkel do not cure the deficiency.

The examiner respectfully disagrees. First it is noted that the applicants' arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

Further, in par. [0189] Verbeke discloses "matching an XML representation of a query ... with representations of the responses that can be provided by each peer".

Verbeke's par. [0180] makes it clear that the "responses that can be provided by each peer" include "code [and] data". Accordingly, those of ordinary skill in the art would have recognized that Verbeke discloses analyzing ("matching an XML representation ... with responses") and retrieving source code (par. [0180] "access ... code, data") which matches a pattern ("XML representation") and thus meets the broadly recited limitations.

In the first par. on pg. 15, the applicants state:

The cited reference of Greenfeld includes "a lexical scanner that separates the source code into tokens", "a parser for determining valid syntactic language", and "a systems analyzer which extracts the programming semantics." There is no reference to indexing source code to be found in Greenfeld. Verbeke, Kataoka, and Ekkel do not cure the deficiency.

The examiner respectfully disagrees. For example, col. 5, lines 3-8 teach a "symbol table subsystem 32 provides a dictionary of symbols found in the source code". Those of ordinary skill in the art would reasonably interpret this "dictionary" as the broadly claimed "index".

#### ***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

**Claims 10-14, 23, and 32-33 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.**

Regarding claims 10, 23, and 32, Applicant defines a system for sharing source code over a network, comprising: a code pattern classifier for analyzing source code generated on a sharing node in the network to identify a set of code patterns, assigning a category to the source code based on the identified set of code patterns and a source code indexer for selectively indexing the source code, dependent code type, and an associated code type. Applicant has defined the systems and their components as merely software and therefore non-statutory as not fitting a statutory category of invention. The systems would have to include a physical component or be stored on a storage medium to be statutory.

The rejection of the base claim are incorporated into their dependent claims.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**Claims 10-12, 14-17, 19 and 23-24 are rejected under 35 U.S.C. 103(a) as being obvious over Verbeke (U.S. Pub 2004/0098447) in view of Greenfeld (U.S. Pat 4931928) and further in view of Kataoka. (U.S. Pat 5,862,382).**

**With respect to claims 10 and 15, Verbeke discloses a system for sharing source code over a network, comprising:**

at least one computer device (e.g. Fig. 2A, Peer node 2003A);

an Integrated Development Environment (IDE), wherein the IDE includes classification-based navigation and content dependent searching for code patterns (par. [0189] "Services ... may be used to facilitate application development");

a code pattern classifier for analyzing source code generated on a sharing node in the network to identify a set of code patterns,(Col 15:0180-0181, "...that share a common set of interests...and access any computer content(code, data, applications,...")

and for assigning at least one predetermined category to the source code based on the identified set of code patterns, wherein code pattern information that is based on the analysis and assignment is stored in a directory;(Col 17:0188-1089, "...indexing, directory...")

but does not disclose and a source code indexer for selectively indexing the source code, a dependent code type, and an associated code type, wherein the source code indexer comprises:

a programming language recognizer for recognizing at least one programming language of the source code;

a selective code content indexer for indexing relevant portions of the source code based on the at least one programming language;

Greenfeld discloses a source code indexer for selectively indexing the source code, a dependent code type, and an associated code type,(Col 4:58-67, "...include a lexical

scanner that separates the source code file into tokens appropriate to the target programming language...") wherein the source code indexer comprises:

    a programming language recognizer for recognizing at least one programming language of the source code;(Col 4:60-67, "...parser for determining valid syntactic structure according to the target programming language...")

    a selective code content indexer for indexing relevant portions of the source code based on the at least one programming language;(Col 4:63-67, "...the parser calls a semantics analyzer which extracts the programming semantics constructs of interest...") in an analogous system for the purpose of extracting programming semantics information during computer source code analysis.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to selectively index source code according to the programming language and code content of the source code shared over the network.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to extract programming semantics information during computer source code analysis.

Neither Verbeke nor Greenfeld disclose a dependency graph indexer for recognizing and indexing a graph of source code dependencies corresponding to the source code;

a code type hierarchy recognizer for recognizing a code type hierarchy associated with the source code; and

an associated code type indexer for indexing a set of code types from the code type hierarchy that is associated with the source code.

Kataoka discloses a dependency graph indexer for recognizing and indexing a graph of source code dependencies corresponding to the source code;(Col 5:59-67, "...the relation extracting means extracts relation information...")

a code type hierarchy recognizer for recognizing a code type hierarchy associated with the source code;(Col 6:17-33, "...structure r1 in which relation Ids for each position are stored...") and

an associated code type indexer for indexing a set of code types from the code type hierarchy that is associated with the source code.(Col 6:17-33 "... a group ID which is a key indicating a group of data associated with each relation in the position...") in an analogous system for the purpose of indexing source code dependencies and hierarchies according the programming language amongst the shared source code.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to selectively index source code dependencies and hierarchies of the source code shared over the network.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to index additional source code information for analysis.

**With respect to claims 11 and 16,** the rejection of claims 10 and 15 are incorporated respectively and further, Verbeke discloses further comprising notifying a set of other nodes in the network of the availability of the source code. (Col 16:0181-0183, "...peer management functions including access control,...")

**With respect to claims 12 and 17,** the rejection of claims 10 and 15 are incorporated respectively and further, Verbeke discloses that the code pattern information comprises the set of code patterns, the at least one category and an identity of the sharing node. (Col 15:0180-0181, "...that share a common set of interests...and access any computer content(code, data, applications,...")

**With respect to claims 14 and 19,** the rejection of claims 10 and 15 are incorporated respectively and further, Verbeke discloses that the accessing step comprises: analyzing working code on the receiving node to identify a context of the working code, and assigning at least one predetermined category to the working code based on the identified context; (Col 17:0188-1089, "...indexing, directory...")

querying the directory to determine a location of the source code; (Col 52:0642-0645, "...query handler...")

and retrieving the source code from the sharing node using the determined location. (Col 17:0189, "...file sharing,...")

**With respect to claims 23 and 24**, Verbeke discloses a system for analyzing source code for sharing over a network comprising:

at least one computer device (e.g. Fig. 2A, Peer node 2003A);  
an Integrated Development Environment (IDE), wherein the IDE includes classification-based navigation and content dependent searching for code patterns (par. [0189] "Services ... may be used to facilitate application development");  
but does not disclose a programming language recognizer or a selective code content indexer.

Greenfeld discloses a programming language recognizer for recognizing at least one programming language of the source code;(Col 4:60-67, "...parser for determining valid syntactic structure according to the target programming language...")

a selective code content indexer for indexing relevant portions of the source code based on the at least one programming language;(Col 4:63-67, "...the parser calls a semantics analyzer which extracts the programming semantics constructs of interest...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to selectively index source code according to the programming language and code content of the source code shared over the network.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to extract programming semantics information during computer source code analysis.

Neither Verbeke nor Greenfeld disclose a dependency graph indexer for recognizing and indexing a graph of source code dependencies corresponding to the source code;

a code type hierarchy recognizer for recognizing a code type hierarchy associated with the source code; and

an associated code type indexer for indexing a set of code types from the code type hierarchy that is associated with the source code.

Kataoka discloses a dependency graph indexer for recognizing and indexing a graph of source code dependencies corresponding to the source code;(Col 5:59-67, "...the relation extracting means extracts relation information...")

a code type hierarchy recognizer for recognizing a code type hierarchy associated with the source code;(Col 6:17-33, "...structure r1 in which relation Ids for each position are stored...") and

an associated code type indexer for indexing a set of code types from the code type hierarchy that is associated with the source code.(Col 6:17-33 "... a group ID which is a key indicating a group of data associated with each relation in the position...") in an analogous system for the purpose of indexing source code dependencies and hierarchies according the programming language amongst the shared source code.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to selectively index source code dependencies and hierarchies of the source code shared over the network.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to index additional source code information for analysis.

**Claims 32-33 and 36-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Verbeke. (U.S. Pub 2004/0098447) in view of Ekkel. (U.S. PGPUB 2003/0088571) further in view of Greenfeld (U.S. Pat 4,931,928) and further in view of Kataoka. (U.S. Pat 5,862,382)**

**With respect to claims 32 and 36,** Verbeke discloses a system for accessing source code shared over a network, (e.g. See Figs. 2A-2D and related text) comprising:  
at least one computer device (e.g. Fig. 2A, Peer node 2003A);

an Integrated Development Environment (IDE), wherein the IDE includes classification-based navigation and content dependent searching for code patterns (par. [0189] "Services ... may be used to facilitate application development");

a context classifier for analyzing working code on the receiving node to identify a context of the working code, and for assigning at least one predetermined category to the working code based on the identified context; (Col 17:0188-1089, "...indexing, directory...")

a query generator for querying a directory using the at least one predetermined category assigned to the working code to identify at least one predetermined category assigned to source code that is relevant to the working code; (Col 52:0642-0645, "...query handler...")

a code pattern requestor for retrieving the source code from a sharing node in the network to the receiving node based on the at least one predetermined category assigned to the source code, (Col 15:0180-0181, "...that share a common set of interests...and access any computer content(code, data, applications,...")

a code pattern classifier for analyzing the source code on the sharing node to identify a set of code patterns, (Col 15:0180-0181, "...that share a common set of interests...and access any computer content(code, data, applications,...")

and for assigning the at least one predetermined category to the source code based on the identified set of code patterns, wherein code pattern information that is based on the analysis and assignment is stored in the directory; (Col 17:0188-1089, "...indexing, directory...")

but does not disclose the code pattern requestor querying a search engine corresponding to the sharing node to retrieve the source code.

Ekkel discloses the code pattern requestor querying a search engine corresponding to the sharing node to retrieve the source code.(Col 1:0010, "...peer-to-peer software applications...") in an analogous system for the purpose of providing a system for securable access to a collection of data across a peer-to-peer data network.(Col 1:0014)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include the code pattern requestor querying a search engine corresponding to the sharing node to retrieve the source code to Verbeke's invention.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide a system for securable access to a collection of data across a peer-to-peer data network.(Col 1:0014)

Neither Verbeke nor Ekkel disclose a source code indexer for selectively indexing the source code, a dependent code type, and an associated code type, wherein the source code indexer comprises:

a programming language recognizer for recognizing at least one programming language of the source code; a selective code content indexer for indexing relevant portions of the source code based on the at least one programming language;

Greenfeld discloses a source code indexer for selectively indexing the source code, a dependent code type, and an associated code type,(Col 4:58-67, "...include a lexical scanner that separates the source code file into tokens appropriate to the target programming language...") wherein the source code indexer comprises:

    a programming language recognizer for recognizing at least one programming language of the source code;(Col 4:60-67, "...parser for determining valid syntactic structure according to the target programming language...")

    a selective code content indexer for indexing relevant portions of the source code based on the at least one programming language;(Col 4:63-67, "...the parser calls a semantics analyzer which extracts the programming semantics constructs of interest...") in an analogous system for the purpose of extracting programming semantics information during computer source code analysis.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to selectively index source code according to the programming language and code content of the source code shared over the network.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to extract programming semantics information during computer source code analysis.

Neither Verbeke nor Ekkel nor Greenfeld disclose a dependency graph indexer for recognizing and indexing a graph of source code dependencies corresponding to the source code;

a code type hierarchy recognizer for recognizing a code type hierarchy associated with the source code; and

an associated code type indexer for indexing a set of code types from the code type hierarchy that is associated with the source code.

Kataoka discloses a dependency graph indexer for recognizing and indexing a graph of source code dependencies corresponding to the source code;(Col 5:59-67, "...the relation extracting means extracts relation information...")

a code type hierarchy recognizer for recognizing a code type hierarchy associated with the source code;(Col 6:17-33, "...structure r1 in which relation Ids for each position are stored...") and

an associated code type indexer for indexing a set of code types from the code type hierarchy that is associated with the source code.(Col 6:17-33 "... a group ID which is a key indicating a group of data associated with each relation in the position...") in an

analogous system for the purpose of indexing source code dependencies and hierarchies according the programming language amongst the shared source code.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to selectively index additional information, i.e. source code dependencies and hierarchies of the source code shared over the network.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to index the additional information.

**With respect to claims 33 and 37,** the rejection of claims 32 and 36 are incorporated respectively and further, Verbeke discloses further comprising a category selector for receiving a response to the query from the directory, (Col 52:0642-0645, "...query handler...") wherein the response includes the at least one predetermined category assigned to the source code, and for selecting the at least one predetermined category assigned to the source code. (Col 17:0188-1089, "...indexing, directory...")

### ***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/  
Jason Mitchell  
10/22/08

/Lewis A. Bullock, Jr./  
Supervisory Patent Examiner, Art Unit 2193